REMARKS

The Non-Formal Office Action, mailed May 1, 2008, considered claims 1-37. Claims 1-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable by Goldberg et al. (US 20040013112), hereinafter Goldberg, in view of Wilson et al. (US 7159119), hereinafter Wilson.

By this response, claims 1, 3-4, 6-7, 9-13, 15, 17-18, 20-23, 25, 27-30, 32-33, & 35-36 are amended while claim 2 is canceled. Claims 1 and 3-37 remain pending of which claims 1, 12, 22, and 29 are independent. Support for the new amendments is found throughout the specification, including, but not limited to the disclosure found in paragraphs [0011] - [0016], and [0030] - [0031].

The present invention is generally directed to embodiments for implementing a method whereby a server can defend itself against denial of service ("DoS") attacks from remote entities without significantly affecting performance. In summary, a server using the embodiments as claimed utilizes a first table of verified remote entities, together with a first hashing function, and a second table of unverified remote entities, together with a second hashing function. The dual tables and hashing functions allow the server to use the first, less computationally intensive yet less secure, hashing function when it encounters packets from a verified entity and the second, more computationally intensive yet more secure, hashing function only when it encounters packets from an unverified entity. Doing so protects the server from a DoS attack by preventing it from adding state information for malicious packets—designed to all comprise the same hash—to an identical index in a single state information table. To do so would cause the server to lose the benefits of a quick hash-based lookup from the state information table, and instead cause it to slowly iterate over the state information for all packets at the hash index for each malicious packet received.

Claim 1 recites a corresponding method in which the server first receives a packet of data containing connection identifier ("CID") information from a remote entity. The server then executes a first hash function on the CID information, generating a first hash which identifies an index in a first table of verified remote entities, the first table storing state information for the remote entity. Each index in the first table of verified remote entities contains state information for each of a plurality of packets comprising the first hash. If an entry for the remote entity exists in the first table of verified remote entities, standard data transport protocol proceeds. Otherwise, the server executes a second cryptographically secure hash function, generating a second hash which identifies an index in a second table of unverified remote entities, the second table storing state information for the remote entity. The second hash function, being cryptographically secure, is more computationally intensive than the first hash function, but with the benefit that it generates a less predicable second hash—decreasing the probability that a plurality of packets comprise the second hash. If an entry does exist in the second table of unverified remote entities, and if the packet contains secret information previously supplied by the server to the remote entity, the server adds the remote entity to the first table of verified entities so that subsequent packets from the remote entity use the first hashing function. Alternately, if an entry does not exist in the second table of unverified remote entities and if the server is a listener, the server can add the remote entity to the second table of unverified remote entities and send the remote entity secret information for later verification.

Claim 22 is directed to a corresponding computer program product comprising computer readable storage media storing computer executable instructions for implementing the method of claim 1.

The other independent claims 12 and 29 are directed to method and computer program product embodiments corresponding to the embodiments recited in claims 1 and 22. However, claims 12 and 29 are recited in functional 'step for' language, as opposed to the non-functional 'acts of' language used in claims 1 and 22.

Claims 1-37 were rejected as being obvious in view of the combination of Goldberg and Wilson. In view of the current amendments, however, Applicant submits that these references fail to teach or suggest each limitation of the independent claims and corresponding dependent claims.

Goldberg is generally directed to embodiments for implementing a firewall or packet filter in a small device such as a PDA or a cellular phone. Goldberg teaches that these small devices have limited computational ability, but that they can quickly process packet information by utilizing a session database using a hash table to index packet data. The hash table facilitates quick lookups of the session information for a packet. Goldberg teaches that a hashing function processes data from a received packet, yielding a hash, and that the hash table uses this hash to index the packet's session information (See [0071]-[0073] of the cited art, for example). The hashing function can generate an identical hash for differing data from a plurality of packets, thus the session database can store multiple entries at the hash index in a doubly-linked list (See [0074] and [0084] of the cited art, for example). Goldberg further teaches that two hashing functions can be used, the first when full socket information is known, and the second when only partial socket information is known. The second hashing function allows temporary openings in the firewall, facilitating dynamic filtering (See [0014] and [0085] of the cited art, for example).

It is noted, however, that the set of hashing functions provided in Goldberg are quite distinct from the set of hashing functions provided by the present invention. Goldberg's first hashing function is somewhat related to the first hashing function recited in the pending claims and which is used to generate hashes which are used as indices in a lookup table containing session information. (See [0071]-[0073] if the cited art, for example). However, it is noted that Goldberg's second hashing function has nothing to do with preventing a plurality of packets from comprising an identical hash, as recited in the pending claims and particularly as recited in combination with the other recited claim elements, such as, but not limited to the limitations that require that the second hashing function be more computationally intensive and more cryptographically secure the first hashing function.

Wilson also fails to teach or suggest any embodiment for using two different hashing functions that are used to prevent a DoS attack, particularly wherein the second hashing function is more computationally intensive and more cryptographically secure than the first hashing function.

Accordingly, it is clear that the pending claims are distinguished from the cited art for at least the foregoing reasons. It will also be noted that the claims are even further distinguished from the cited art for other reasons as well. For example, Goldberg and Wilson also fail to teach or suggest any embodiment that includes multiple lookup tables. According to the present invention, two lookup tables exist, both of which are used to index packet session information. Two lookup tables are necessary to prevent DoS attacks from unverified entities while still maintaining quick lookups for verified entities (see [0011] – [0016] of the application as originally filed, for example).

Wilson was cited as purportedly disclosing multiple tables. However, Wilson's tables are notably distinguished from the first verified and the second unverified tables recited in the pending claims. Wilson does teach that a system can utilize and search multiple lookup tables consecutively to determine whether a search key exists, and if the key does exist, to determine in which category the key exists.[1] However, Wilson's tables are never used to determine whether a remote entity is verified or unverified as required by the pending claims and are, therefore, quite distinct from the present invention as claimed. Instead, Wilson teaches that two lookup tables can be used to divide categories of information, but fails to teach that any un-hashed data is stored at the indices of the tables—teaching instead that data is pre-processed and that only a hash of the data is contained in the lookup table. The present invention requires later access to the data placed within the lookup table, and thus bears little relation to the lookup tables taught by Wilson.

Notably, like Wilson, Goldberg also fails to disclose multiple tables, including a first verified table and a second unverified table, as claimed. Instead, Goldberg discloses only a single lookup table which is used to speed access to session information. Goldberg clearly fails to teach or suggest the use or presence of two lookup tables for use in prevent a DoS attack, and particularly in the manner claimed by the present invention.

In view of the foregoing, Applicant respectfully submits that all the rejections to the independent claims are now moot and that the independent claims are now allowable over the cited art, such that any of the remaining rejections and assertions made, particularly with respect to all of the dependent claims, do not need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice, and particularly with regard to the dependent claims.[2]

[1] Wilson is generally directed to embodiments for securely searching sensitive content by implementing a database which does not contain a copy of the content being searched. Wilson teaches that although the content does not exist in the database, the database can reveal whether a search key, or permutations of it, exists in the content. Specifically, Wilson provides a means to adhere to statutory restrictions by implementing a database used to determine if a standardized mailing address is valid without presenting a risk of revealing all valid addresses. Wilson pre-processes the content (e.g. all valid mailing addresses), first by dividing it into categories (e.g. "business addresses" and "individual addresses"), and then by executing a hashing function on the content. Only a hash of a piece of content (e.g. an address), not the content itself, is included in a lookup table corresponding to the respective category (e.g. "business addresses" or "individual addresses").

[2] Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting any official notice taken. Furthermore, although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at 801-533-9800.

Dated this 1st day of August, 2008.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
JENS C. JENKINS
Registration No. 44,803
Attorneys for Applicant
Customer No. 47973

RDN:JCJ:jml:laf
1766187_1